

DECISION MAKING

Application of if-then, if-then-else, switch (default, break).

Q. What is a statement?

Ans: Statements are the instructions given to the computer to perform any kind of action, as data movements, making decision or repeating action. Statements form the smallest executable unit and terminated with semi-colon.

Q. What are the three constructs that govern statement flow?

Ans: The three constructs that governs statement flow are: Sequence, Selection and Iteration constructs.

Q. What is a selection/conditional statement? Which selection statements does Java provides?

Ans: A selection statement is the one that is used to decide which statement should

be execute next. This decision is based upon a test condition. The selection statements provided by Java are: if-else and switch. The conditional operator ?: can also be used to take simple decision.

Q. What is an ‘if’ statement? Explain with an example.

Ans: the ‘if’ statement helps in selecting one alternative out of the two. The execution of ‘if’ statement starts with the evaluation of condition. The ‘if’ statement therefore helps the programmer to test for the condition. General form of ‘if’ statement.

```
if(expression) statement
```

```
if(marks>=80)
```

```
System.out.println(“Grade A”);
```

Q. What is the significance of a test-condition in a if statement?

Ans: It is the test condition of an if statement that decides whether the code associated with the if part or the one

associated with the else part should be executed. The former is executed if the test condition evaluates to true and the latter works if the condition evaluates to false.

Q. Write one advantage and one disadvantage of using ?: in place of an if.

Ans: Advantage: It leads to a more compact program. Disadvantage: Nested ?: becomes difficult to understand or manage.

Q. What do you understand by nested 'if' statements?

OR

Q. Explain with an example the if-else-if construct.

Ans: A nested 'if' is an statement that has another 'if' in its body or in it's appearance. It takes the following general form.

```
if(ch>='A')  
{  
if(ch<='Z')  
++upcase;
```

```
else
++other;
}
```

Q. What is the problem of dangling-else? When does it arise? What is the default dangling-else matching and how it be overridden?

Ans: The nested if-else statement introduces a source of potential ambiguity referred to as dangling-else problem. This problem arises when in a nested if statement, number of if's is more than the number of else clause. The question then arises, with which if does the additional else clause properly match. For Example

```
if(ch>='A')
if(ch<='Z')
++upcase;
else
++other;
```

The indentation in the above code fragment indicates that programmer wants the else to be with the outer if. However

Java matches an else with the preceding unmatched if. One method for over-riding the default dangling-else matching is to place the last occurring unmatched if in a compound statement, as it is shown below.

```
if(ch>='A')
{
if(ch<='Z')
++upcase;
}
else
++other;
```

Q. Compare and contrast IF with ?:

Ans: (i) Compare to IF sequence, ?: offer more concise, clean and compact code, but it is less obvious as compared to IF. (ii) Another difference is that the conditional operator ?: produces an expression, and hence a single value can be assigned, for larger expression If is more flexible. (iii) When ?: operator is used in its nested form, it becomes complex and difficult to understand.

Q. What is a switch statement? How is a switch statement executed?

Ans: Switch statement successively tests the value of an expression against a set of integers or character constants. When a match is found, the statements associated with the constants are executed. The syntax `switch(expression)`

```
{  
case constants : statements; break;  
case constants : statements; break;  
}
```

The expression is evaluated and its values are matched against the value of the constants specified in the case statements. When a match is found, the statements sequence associated with that case is executed until the break statement or the end of switch statement is reached.

Q. What is the significance of break statement in a switch statement?

Ans: In switch statement when a match is found the statement sequence of that case

is executed until a 'break' statement is found or the end of switch is reached, when a 'break' statement is found program execution jumps to the line of code following the switch statement.

Q. What is a control variable in a switch case?

Ans: A control variable in switch case is one which guides the control to jump on a specified case. e.g. switch(x), here 'x' is the control variable.

Q. What is a "fall through"?

Ans: The term "fall through" refers to the way the switch statement executes its various case sections. Every statement that follows the selected case section will be executed unless a break statement is encountered.

Q. What is the effect of absence of break in a switch statement?

Ans: Absence of break statement in a switch statement leads to situation called

“fall through” where once a matching case is found the subsequent case blocks are executed unconditionally

Q. Write one limitation and one advantage of switch statement?

Ans: Advantage: More efficient in case a value is to be tested against a set of constants. Disadvantage: switch can test only for equality, so for the rest of comparisons one needs to use if-else.

Q. Discuss when does an if statement prove more advantageous than switch statement.

Ans: In the following case if statement proves to be more advantage over switch statement: (i) When a range of values need to be tested for. (ii) When relation between multiple variables needs to be tested. (iii) When multiple conditions need to be tested. (iv) When expressions having a data type other than integer or character need to be tested.

Q. When does switch statement prove more advantageous over an if statement?

Ans: The switch statement is more advantageous than the if statement when the test expression whose data type is either of byte, short, character, integer or long is to be tested against a set of constants. The reason being that the switch statement evaluates the expression once whereas the equivalent if statement evaluates the expression repeatedly.

Q. Explain, with the help of an example, the purpose of default in a switch statement.

Ans: The default section is an optional part of the switch statement and the statement written under default clause are executed when no matching case is found.

```
switch(n)
```

```
{
```

```
case 1: System.out.println("Sunday");
```

```
break;
```

```
case 2: System.out.println("Monday");
```

```
break;
case 3: System.out.println("Tuesday");
break;
case 4: System.out.println("Wednesday");
break;
case 5: System.out.println("Thursday");
break;
case 6: System.out.println("Friday");
break;
case 7: System.out.println("Saturday");
break;
default : System.out.println("Invalid
Input");
}
```

Q. Differentiate between if and switch statements.

Ans: Both are used as a selection statements, there are some difference in their operations. (i) switch can only test for equality, where as if can evaluate a relational or logical expression. (ii) it statement can handle ranges , where as switch case level must be a single value.

(iii) if statement can handle floating point test also, where as the switch case labels must be an integer or character.